Reflexionsjournal

Software Task

Deadline: two weeks after handout

*ACT: erstmal ins Deutsche übersetzen. Dabei sieht (und versteht) man mehr Details und das spätere "überfliegen" ist effizienter. (Für mich zumindest)*

Bitte führen Sie eine Anforderungsanalyse für die unten beschriebene Anforderung "Simple Graph Calculator and Plotter" durch. Ermitteln Sie relevante Teilaufgaben und nennen Sie weitere relevante Aspekte einer Anforderungsanalyse.

Bitte erstellen Sie ein geeignetes Softwarekonzept für die genannte Anforderung und dokumentieren Sie es in der von Ihnen gewünschten Form. Versuchen Sie dabei, bekannte und bewährte Methoden zu verwenden.

3. [Optional] Erstellen Sie einen Prototyp in C# .Net 4.8 und WPF. Liefern Sie ausführbare Dateien und Quellcode. Verwenden Sie keine eingeschränkten Softwarekomponenten oder Bibliotheken für kommerzielle Zwecke. Beschreiben Sie mögliche Zusammenhänge zwischen dem Prototyp und einem endgültigen Softwareprodukt.

HINTERGRUND & MOTIVATION

- Einfache Eingabe von Funktionsparametern für definierte Funktionen, um die Auswirkungen und Veränderungen in einer Grafik zu sehen.

- Der Benutzer kann die Auswirkungen der Parameter leicht erkennen.

FEATURE HIGH-LEVEL DESCRIPTION

EINFACHER GRAPHENRECHNER & PLOTTER

Windows-Anwendung zur Eingabe der Parameter für trigonometrische Funktionen (sin und cos) und die Sinc-Funktion. Die Funktionen sollen in 2D visualisiert werden. Der Bereich muss anpassbar sein. Der Typ der Funktion soll wählbar sein.

SPECIFICATION DESCRIPTION

FOLGENDE ERWEITERTE FUNKTIONEN

- Nach dem Öffnen der Anwendung sollten die letzten Parameter sichtbar sein.

- Die Visualisierung sollte im Vektorformat speicherbar sein

*Die Anforderungsanalyse ist typischerweise nicht die Hauptaufgabe eines Entwicklers, sondern eher die eines Business Analysts oder Product Owners. Da ich die letzten Jahre als beides nicht tätig war, fehlen mir die Erfahrungen diese, in einem bestimmten Format abzugeben. Stattdessen werde ich alles so wiedergeben, das ich damit etwas anfangen kann um einen Prototypen zu entwickeln.*

*OK, was haben wir denn alles....*

**Grobe Zusammenfassung der Aufgabe:**

Windows Desktop Anwendung mit C# WPF (und MVVM)

Fenster mit a) einer Auswahl der Funktion (Combobox)

b) entprechende Eingabefelder für die Parameter und c)

einem Bereich zur AUsgabe des Graphen.

*Jetzt ist die Frage die ich mir stelle "Soll das erweiterbar sein?" d.h sollen später auch andere Funktionen unterstützt werden und muss die Parametereingabe daher als dynamische Liste erstellt werden. Ich gehe jetzt erst mal davon aus, das nur das umgesetzt werden soll, was beschrieben wurde. In der Praxis würde ich eine Rückfrage stellen. oder wenn ich selbst programmiere, schauen was mit geringem Aufwand möglich ist.*

**Grobe Teilaufgaben:**

- Hauptfenster wie oben beschrieben mit ViewModel.

- Komponente zum Berechnen der Funktion

- Komponente zum Zeichnen des Graphen.

- Komponente für die Persistierung der Eingabefelder.

- Komponente für die Speicherung des Graphen.

**Anforderungen überprüfen:**

- Einfache Eingabe von Funktionsparametern für definierte Funktionen, um die Auswirkungen und Veränderungen in einer Grafik zu sehen.

**CHECK**: Window wie oben beschrieben

- Der Benutzer kann die Auswirkungen der Parameter leicht erkennen.

**CHECK**: Nach (oder wärend) der Eingabe der einzelnen Parameter wird die neue Funktion gezeichnet, die alte bleibt erhalten bis die Eingabe vollständig ist (Apply-Button?)

- Windows-Anwendung zur Eingabe der Parameter für trigonometrische Funktionen (sin und cos) und die Sinc-Funktion.

**CHECK**: wie oben beschrieben

- Die Funktionen sollen in 2D visualisiert werden.

**CHECK**: Canvas and Path (einfachste Umsetzung) oder Fremdkomponente (Diagramm-Komponente)

- Der Bereich muss anpassbar sein.

**CHECK**: zusätzliche Eingabefelder für Min, Max, Schrittweite für x/y Achse

- Der Typ der Funktion soll wählbar sein.

**CHECK**: Combobox (als einfachste Umsetzung) OPTIONAL Karussel mit Vorschaugrafik als "Luxus"-Variante

- Nach dem Öffnen der Anwendung sollten die letzten Parameter sichtbar sein.

**CHECK**: Nach öffnen der Anwendung wird die Komponente zur Persistierung verwendet um im ViewModel die Werte zu setzen. Nach Berechnung einer Funktion, oder beim Beenden des Programms werden die aktuelle Werte gespeichert.

- Die Visualisierung sollte im Vektorformat speicherbar sein

**CHECK**: XAML ist ein Vektorformat, Der Path könnte so gespeichert werden wie er auch gezeichnet wird, als XAML. Es ist nicht spezifiziert was genau gespeichert werden soll, Das ganze Diagramm (Canvas und Path) oder nur der Path. OPTIONAL Wird ein spezielles Vectorformat (z.B. SVG) gewünscht, entsprechende Konverter erstellen oder exterrne Komponenten verwenden.

*So bis hierhin geht das noch alles rein theoretisch. Jetzt würde ich mir jede einzelne Komponente vorknöpfen und schauen was technisch machbar ist und wie man das umsetzen könnte. Wenn man das in reiner Theorie macht, ohne zwischendurch zu testen, hat man (ich) die Hälfte übersehen. Außerdem verliert man (ich) schneller den Überblick.*

*Es wäre eventuell auch nützlich noch ein paar zusätzliche Augen dazu zu nehmen, um Dinge hinzuzufügen die man bisher übersehen hat.*

*Damit ich weiß wie so was aussehen könnte, lass ich ChatGPT eine [Anforderungsanalyse] und ein [Softwarekonzept] erstellen. Bei späteren Aufgabe sollte man so was natürlich selber machen. Aber ich brauche irgendeinen Startpunkt. Eine Schulung dazu wäre nicht schlecht. Leider fehlt mir die Zeit dazu.*

**Zu den Details:**

- Hauptfenster wie oben beschrieben mit ViewModel.

...

- Komponente zum Berechnen der Funktion

...

- Komponente zum Zeichnen des Graphen.

...

- Komponente für die Persistierung der Eingabefelder.

...

- Komponente für die Speicherung des Graphen.

...

*Da ich ein eher praktisch veranlagter Mensch bin, und mir die Zeit wegläuft, fange ich den Prototypen an. Ich weiß, das ist nicht das exakt das was erwartet wird, aber es ist das was ich kann. Beim Entwickeln habe ich bessere Ideen, als beim trockenem Beschreiben der Aufgaben. Das ist wohl eine Übungssache, an der ich arbeiten muss. Am Ende zählt erstmal, ob der Kunde seine Software in dem geplanten Zeitraum bekommt.*

*Während ich daran arbeite, sehe ich, welche Komponenten komplexer (oder einfacher) sind und passe die Struktur des Projektes entsprechend an. Mein Ziel ist es, eine erweiterbare Software so zu entwickeln, das auch der nächste Softwareentwickler mit dem Code etwas anfangen kann.*

*Nachdem ich den Prototypen mit den grundsätzlichen Funktionen fertig habe und auch die Anforderungsanalyse und Softwarekonzept leicht überarbeitet habe, muss ich resümieren, das ich in theoretischer Softwareentwicklung noch sehr viel Nachholbedarf habe. Aber der Prototyp ist presentier fertig.*